Open|SpeedShop™



Silicon Graphics, Inc.

Open|SpeedShop™ Overview, Status, Upcoming Features

Presented by:

Jack Carter

SGI Core Engineering

April 2006

SGI Proprietary

©2006 Silicon Graphics, Inc. All rights reserved. Silicon Graphics, SGI, the SGI Logo, the SGI cube, and The Source of Innovation and Discovery are registered trademarks and Open|SpeedShop and SGI SpeedShop are trademarks of Silicon Graphics Inc., in the United States and/or other countries worldwide. Linux is a registered trademark of Linus Torvalds in several countries, used with permission by Silicon Graphics, Inc. Qt and the Qt logo are trademarks of Trolltech in Norway, the United States and other countries. IBM is a registered trademarks of International Business Machines Corporation in the United States, other countries, or both. All other trademarks mentioned herein are the property of their respective owners.

Agenda

- Performance Analysis Tools
- Background
- Overview
- Currently Supported Features
- Coming Attractions
- Q&A

Open SpeedShop

sgi

Performance Analysis Tools

- What is a performance analysis tool?
 - A tool for analyzing performance bottlenecks in an application
 - A tool that finds where and why the application is spending the most time
- Who needs a performance analysis tool?
 - Anyone writing or maintaining a performance sensitive application
 - Anyone tuning an application for a customer

sgi

Open|SpeedShop: Background

- Background
 - DOE/NNSA co-funded - as part of ASC PathForward
 - Open Source under GPL and LGPL licenses
 - Implemented Using Open Source Components
 - DPCL – multi-platform cluster capability
 - Dyninst – dynamic instrumentation capability
 - PAPI – multi-platform hardware counter access
 - SQLite – lightweight multi-platform database

Open|SpeedShop: Overview

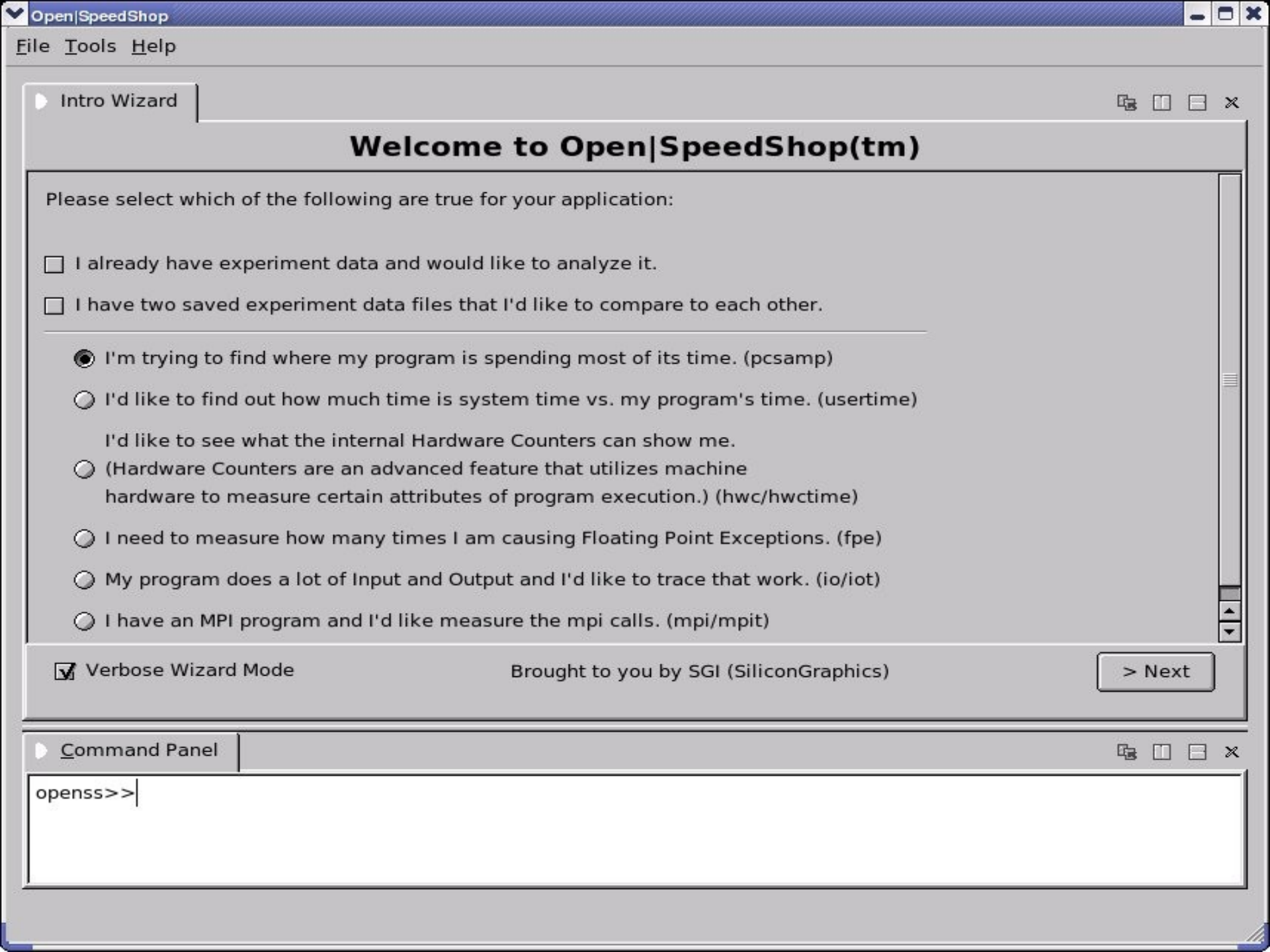
- Overview
 - Performance Analysis Tool for Linux®
 - Minimal perturbation of the application being analyzed
 - Post mortem analysis
 - Cluster Friendly
 - Easy to use
 - Multiple UIs that accommodate different working styles
 - No recompilation of application being analyzed is needed
 - Wizard to help make initial decisions

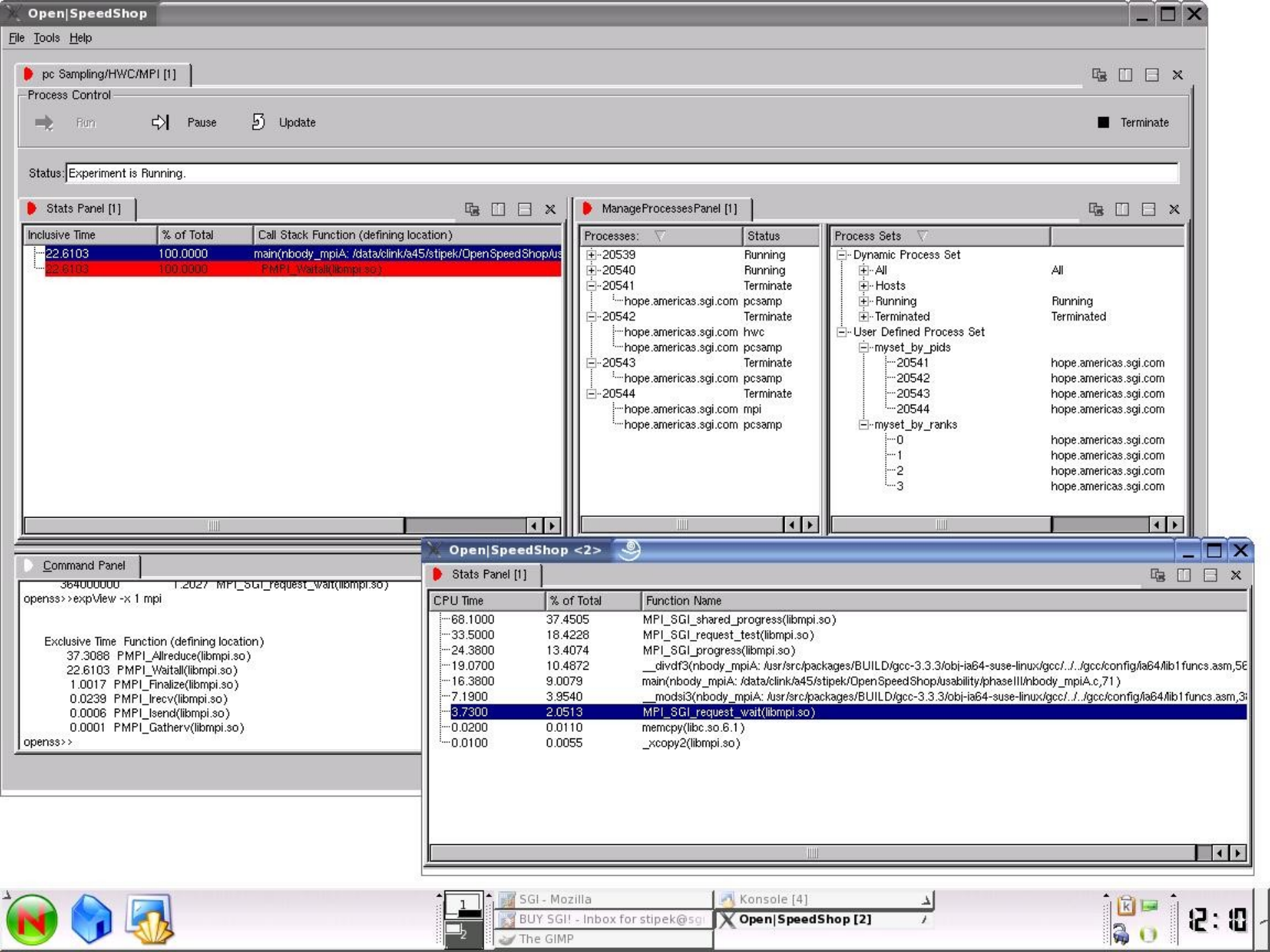
Currently Supported Features

- Three Modes of Operation
 - Batch/Command file
 - Interactive Command Line Interface (CLI)
 - Graphical User Interface (GUI)
- Create your own experiments
- Multiple System Support
 - Single System Image (SSI)
 - Traditional Cluster
 - SGI Supercluster
- Supports various MPI Implementations
 - SGI Message Passing Toolkit
 - MPICH
 - LAMPI
 - OPENMPI
- Scales to large numbers of processes, threads and ranks

Currently Supported Features (2)

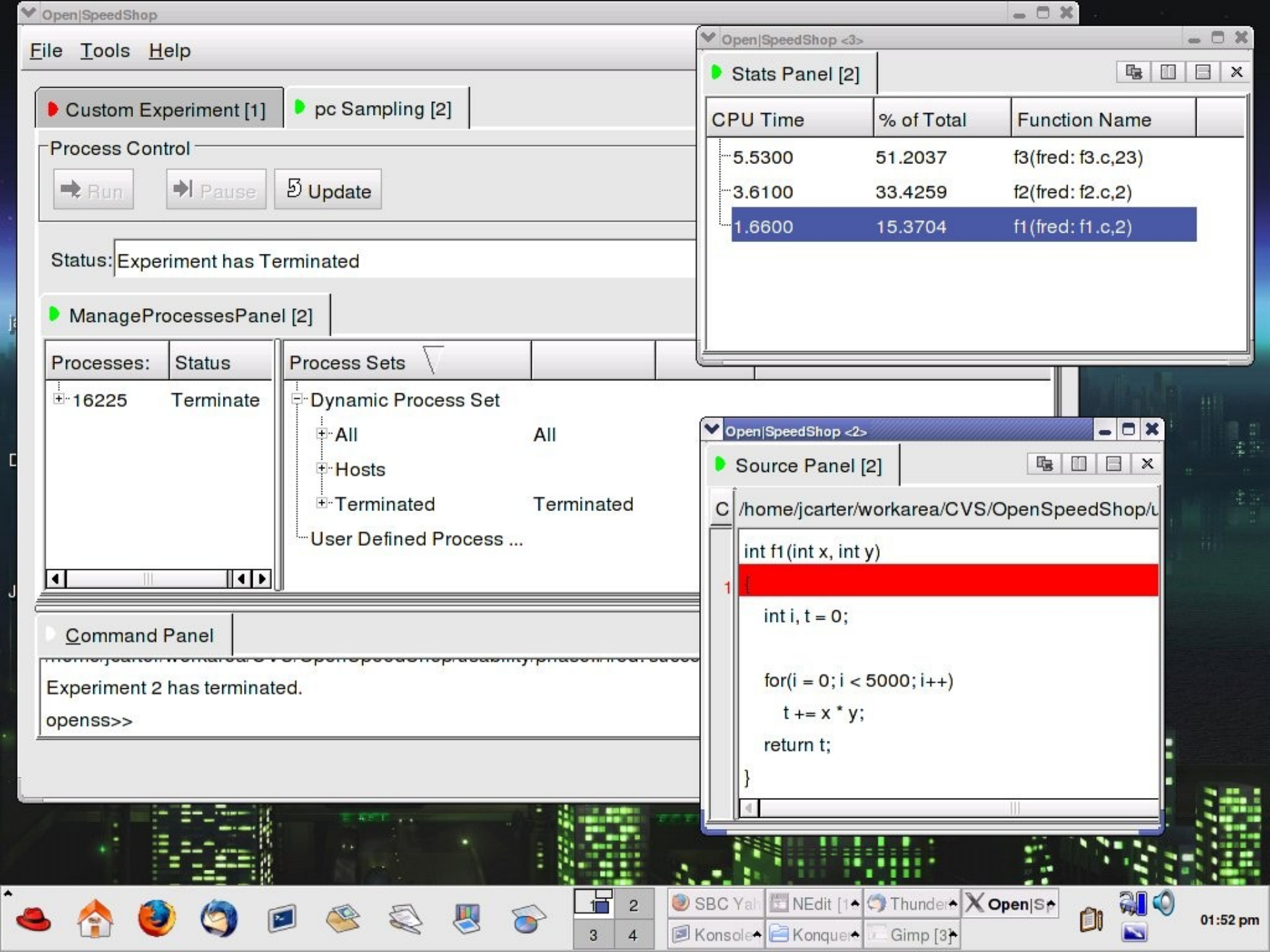
- Attach to running processes, threads and ranks to gather performance data
- Intermediate performance measurement data view and capture
- Save and restore performance experiment information for post experiment performance analysis
- GUI wizard facility and context sensitive help
- GUI Allows multiple windows

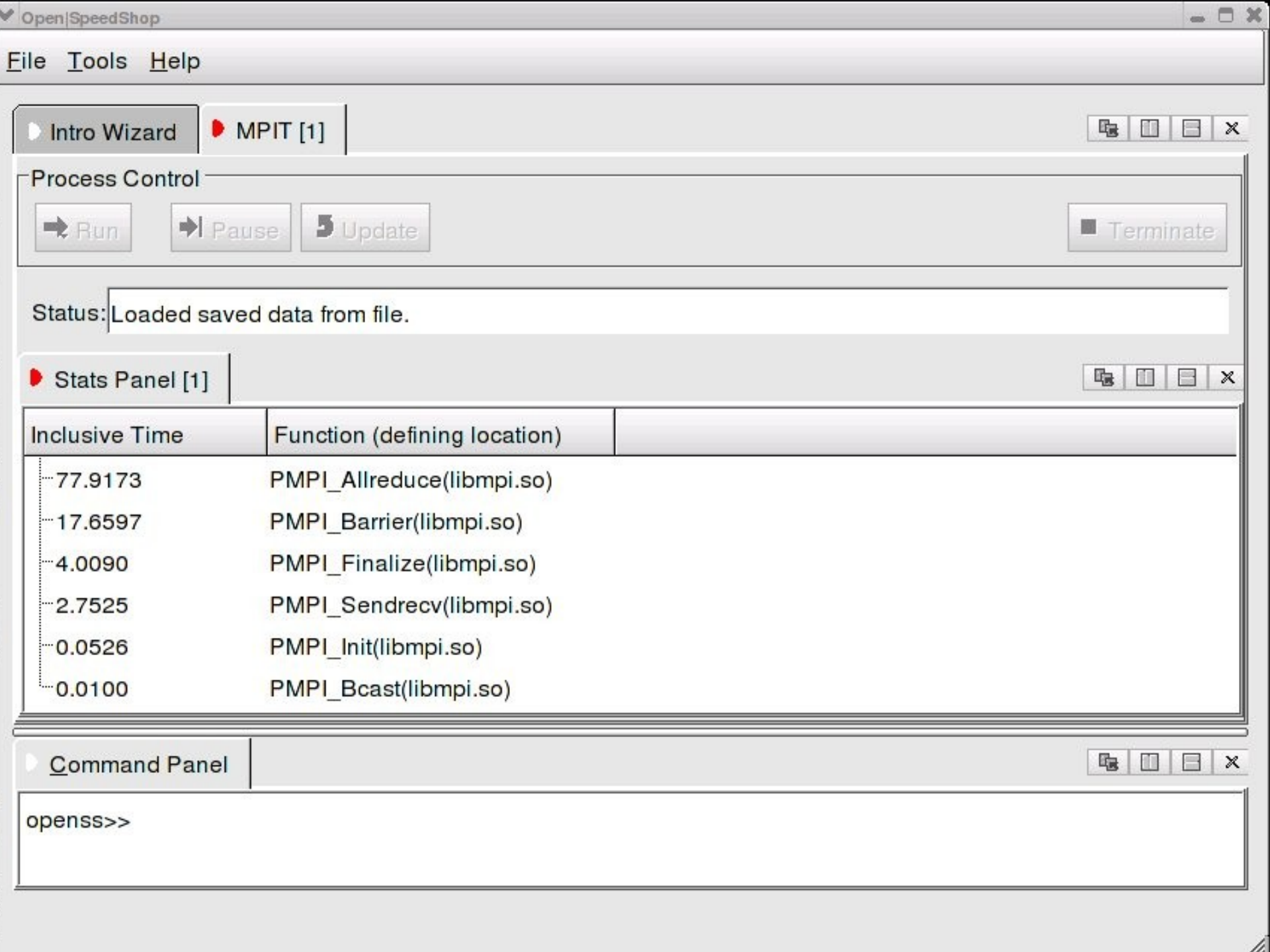

Open|SpeedShop
File Tools Help
Intro Wizard
Welcome to Open|SpeedShop(tm)
Please select which of the following are true for your application:
I already have experiment data and would like to analyze it.
I have two saved experiment data files that I'd like to compare to each other.
I'm trying to find where my program is spending most of its time. (pcsamp)
I'd like to find out how much time is system time vs. my program's time. (usertime)
I'd like to see what the internal Hardware Counters can show me.
(Hardware Counters are an advanced feature that utilizes machine
hardware to measure certain attributes of program execution.) (hwc/hwctime)
I need to measure how many times I am causing Floating Point Exceptions. (fpe)
My program does a lot of Input and Output and I'd like to trace that work. (io/iot)
I have an MPI program and I'd like measure the mpi calls. (mpi/mpit)
Verbose Wizard Mode
Brought to you by SGI (SiliconGraphics)
> Next
Command Panel
openss>>


Open|SpeedShop
File Tools Help
pc Sampling/HWC/MPI [1]
Process Control
Run
Pause
Update
Terminate
Status: Experiment is Running.
Stats Panel [1]
Inclusive Time
% of Total
Call Stack Function (defining location)
22.6103
100.0000
main(nbody_mpiA: /data/clink/a45/stipek/OpenSpeedShop/us
22.6103
100.0000
PMPI_Waitall(libmpi.so)
ManageProcessesPanel [1]
Processes:
Status
20539 Running
20540 Running
20541 Terminate
hope.americas.sgi.com pcsamp
20542 Terminate
hope.americas.sgi.com hwc
hope.americas.sgi.com pcsamp
20543 Terminate
hope.americas.sgi.com pcsamp
20544 Terminate
hope.americas.sgi.com mpi
hope.americas.sgi.com pcsamp
Process Sets
Dynamic Process Set
All All
Hosts
Running Running
Terminated Terminated
User Defined Process Set
myset_by_pids
20541 hope.americas.sgi.com
20542 hope.americas.sgi.com
20543 hope.americas.sgi.com
20544 hope.americas.sgi.com
myset_by_ranks
0 hope.americas.sgi.com
1 hope.americas.sgi.com
2 hope.americas.sgi.com
3 hope.americas.sgi.com
Command Panel
openss>>expview -x 1 mpi
Exclusive Time Function (defining location)
37.3088 PMPI_Allreduce(libmpi.so)
22.6103 PMPI_Waitall(libmpi.so)
1.0017 PMPI_Finalize(libmpi.so)
0.0239 PMPI_Irecv(libmpi.so)
0.0006 PMPI_Isend(libmpi.so)
0.0001 PMPI_Gatherv(libmpi.so)
openss>>
Open|SpeedShop <2>
Stats Panel [1]
CPU Time
% of Total
Function Name
68.1000 37.4505 MPI_SGI_shared_progress(libmpi.so)
33.5000 18.4228 MPI_SGI_request_test(libmpi.so)
24.3800 13.4074 MPI_SGI_progress(libmpi.so)
19.0700 10.4872 __divdf3(nbody_mpiA: /usr/src/packages/BUILD/gcc-3.3.3/obj-ia64-suse-linux/gcc/../../gcc/config/ia64/lib1funcs.asm,58
16.3800 9.0079 main(nbody_mpiA: /data/clink/a45/stipek/OpenSpeedShop/usability/phaseIII/nbody_mpiA.c,71)
7.1900 3.9540 __modsi3(nbody_mpiA: /usr/src/packages/BUILD/gcc-3.3.3/obj-ia64-suse-linux/gcc/../../gcc/config/ia64/lib1funcs.asm,3
3.7300 2.0513 MPI_SGI_request_wait(libmpi.so)
0.0200 0.0110 memcpy(libc.so.6.1)
0.0100 0.0055 _xcopy2(libmpi.so)
SGI - Mozilla
BUY SGI! - Inbox for stipek@sgi
The GIMP
Konsole [4]
Open|SpeedShop [2]
12:40

Currently Supported Features (3)

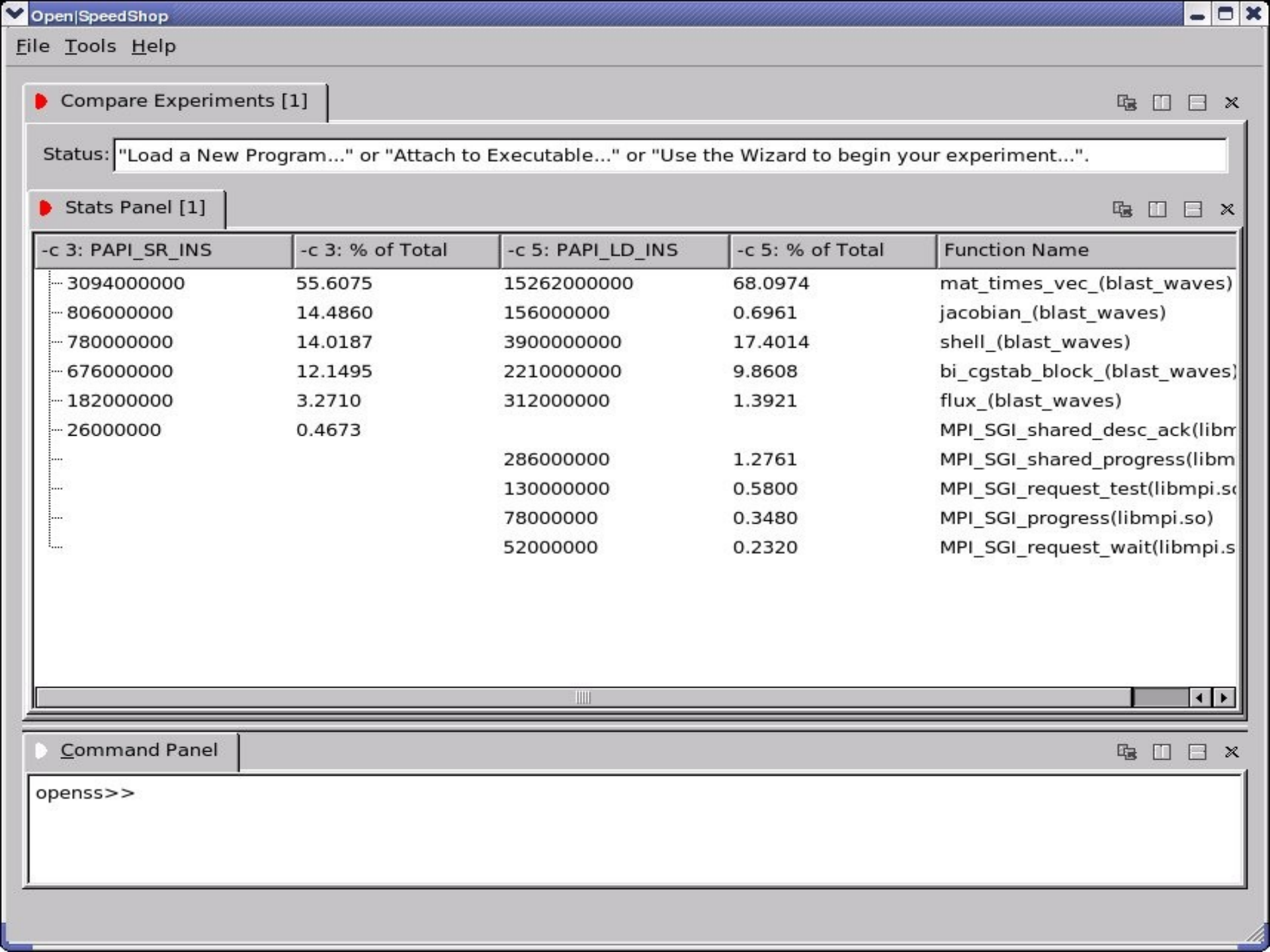
- Basic experiments
 - Instruction Pointer Sampling based on time (pcsamp)
 - Stack Trace Sampling based on time (usertime)
 - Instruction Pointer Sampling based on hardware counter overflow (hwc)
 - Stack Trace Sampling based on hardware counter overflow (hwctime)
 - Input/Output Call Event Tracing (io)
 - Input/Output Call Event Tracing with more detailed information (iot)
 - MPI Call Event Tracing (mpi)
 - MPI Call Event Tracing with more detailed information (mpit)


Open|SpeedShop
File Tools Help
Custom Experiment [1]
pc Sampling [2]
Process Control
Run
Pause
Update
Status: Experiment has Terminated
ManageProcessesPanel [2]
Processes:
Status
16225
Terminate
Process Sets
Dynamic Process Set
All
All
Hosts
Terminated
Terminated
User Defined Process ...
Command Panel
Experiment 2 has terminated.
openss>>
Open|SpeedShop <3>
Stats Panel [2]
CPU Time
% of Total
Function Name
5.5300
51.2037
f3(fred: f3.c,23)
3.6100
33.4259
f2(fred: f2.c,2)
1.6600
15.3704
f1(fred: f1.c,2)
Open|SpeedShop <2>
Source Panel [2]
/home/jcarter/workarea/CVS/OpenSpeedShop/
int f1(int x, int y)
{
int i, t = 0;
for(i = 0; i < 5000; i++)
t += x * y;
return t;
}
01:52 pm


Open|SpeedShop
File Tools Help
Intro Wizard
MPIT [1]
Process Control
Run
Pause
Update
Terminate
Status: Loaded saved data from file.
Stats Panel [1]
Inclusive Time
Function (defining location)
77.9173
PMPI_Allreduce(libmpi.so)
17.6597
PMPI_Barrier(libmpi.so)
4.0090
PMPI_Finalize(libmpi.so)
2.7525
PMPI_Sendrecv(libmpi.so)
0.0526
PMPI_Init(libmpi.so)
0.0100
PMPI_Bcast(libmpi.so)
Command Panel
openss>>

Currently Supported Features (4)

- Custom data comparison views
 - Create columns for your view
 - Compare different runs of same program with different data
 - Compare different hardware counter experiments


Open|SpeedShop
File Tools Help
Compare Experiments [1]
Status: "Load a New Program..." or "Attach to Executable..." or "Use the Wizard to begin your experiment...".
Stats Panel [1]
-c 3: PAPI_SR_INS
-c 3: % of Total
-c 5: PAPI_LD_INS
-c 5: % of Total
Function Name
3094000000 55.6075 15262000000 68.0974 mat_times_vec_(blast_waves)
806000000 14.4860 156000000 0.6961 jacobian_(blast_waves)
780000000 14.0187 3900000000 17.4014 shell_(blast_waves)
676000000 12.1495 2210000000 9.8608 bi_cgstab_block_(blast_waves)
182000000 3.2710 312000000 1.3921 flux_(blast_waves)
26000000 0.4673 MPI_SGI_shared_desc_ack(libm
286000000 1.2761 MPI_SGI_shared_progress(libm
130000000 0.5800 MPI_SGI_request_test(libmpi.s
78000000 0.3480 MPI_SGI_progress(libmpi.so)
52000000 0.2320 MPI_SGI_request_wait(libmpi.s
Command Panel
openss>>

Currently Supported Features (5)

- Python Scripting API module
 - Use Open|SpeedShop features from within a pure python environment
 - Create scripts in a known language
 - Capture data either as a formatted view or as data for further processing

j.py - /home/jcarter/workarea/CVS/OpenSpeedShop/pyscripting/test/

File Edit Search Preferences Shell Macro Windows Help

/home/jcarter/workarea/CVS/OpenSpeedShop/pyscripting/test/j.py 763 bytes L: 3 C: 13

```
#!/usr/bin/python

import openss

try :
    # Build up the experiment marching orders
    my_file = openss.FileList("../../usability/phaseII/fred 900")
    my_exptype = openss.ExpTypeList("pcsamp")
    my_expid = openss.expCreate(my_file,my_exptype)

    openss.expGo()

    # The program runs asynchornously without this
    openss.wait()

    # Describe the view
    my_viewtype = openss.ViewTypeList("pcsamp")
    my_metric_list = openss.MetricList(["time"])

    # Get formatted output
    data1 = openss.expView(my_expid, my_viewtype,my_metric_list)
    openss.dumpInfo(data1)

    # Get plain data output
    data2 = openss.expData(my_expid, my_viewtype,my_metric_list)
    openss.dumpInfo(data2)

except openss.error,message:
    print "ERROR: ", message
```

Coming Attractions

- Floating Point Exception Tracing (fpe)
- Run applications with 4000 processes in reasonable time
- MPI rank support
- Automatic detection of outliers
- Support for gathering per-thread data
- To be provided with SGI ProPack 5

Questions



Resources

- Project Lead & Manager
 - Jim Galarowicz: jeg@sgi.com
- Questions
 - Jack Carter: jcarter@sgi.com
- Live CVS Repository available now
 - http://oss.sgi.com/openspeedshop
- Bugzilla database
 - http://oss.sgi.com/bugzilla (search for OpenSpeedShop)